

FILEID**TPR

H 1

TTTTTTTTT1 PPPPPPPP RRRRRRRR
TTTTTTTTT2 PPPPPPPP RRRRRRRR
TT PP PP RR RR
TT P PPPPPP RRRRRRRR
TT P PPPPPP RRRRRRRR
TT PP RR RR
TT PP RR RR

.....
.....
.....

LL I II II SSSSSSSS
LL I II II SSSSSSSS
LL I II II SS
LL I II II SS
LL I II II SS
LL I II II SSSSSS
LL I II II SSSSSS
LL I II II SS
LL I II II SS
LL I II II SS
LLLLLLLLL1 I II II SSSSSSSS
LLLLLLLLL2 I II II SSSSSSSS

TPF
VO4

1 0001 0 MODULE TPR !
2 0002 0 IDENT = 'V04-000'
3 P 0003 0 XBLISS32[
4 P 0004 0 ADDRESSING_MODE(INTERNAL=LONG_RELATIVE,NONEXTERNAL=LONG_RELATIVE)
5 0005 0]
6 0006 0) =
7 0007 1 BEGIN
8 0008 1 !
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 * ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 * TRANSFERRED.
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 * CORPORATION.
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1 !
32 0032 1 ++
33 0033 1 | FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
34 0034 1 | ABSTRACT: Routines for checking how full the text portion of a page is.
35 0035 1 | ENVIRONMENT: Transportable
36 0036 1 |
37 0037 1 | AUTHOR: R.W.Friday CREATION DATE: May, 1978
38 0038 1 |
39 0039 1 |
40 0040 1 |

TPR
V04-000

Revision History

J 1
16-Sep-1984 01:55:26
14-Sep-1984 13:08:21
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]TPR.BLI;1 Page 2
**F (2)

42 0041 1 %SBTTL 'Revision History'
43 0042 1
44 0043 1 MODIFIED BY:
45 0044 1
46 0045 1 007 KAD00007 Keith Dawson 07-Mar-1983
47 0046 1 Global edit of all modules. Updated module names, idents,
48 0047 1 copyright dates. Changed require files to BLISS library.
49 0048 1
50 0049 1 ---

```
52 0050 1 %SBTTL 'Module Level Declarations'  
53 0051 1  
54 0052 1 !  
55 0053 1 ! TABLE OF CONTENTS:  
56 0054 1 !  
57 0055 1 !  
58 0056 1 FORWARD ROUTINE  
59 0057 1 TPR,  
60 0058 1 TPBÉQL,  
61 0059 1 TPFEQL,  
62 0060 1 TPFFIT,  
63 0061 1 TPFSKP,  
64 0062 1 TPFSIZ;  
65 0063 1  
66 0064 1 ! INCLUDE FILES:  
67 0065 1 !  
68 0066 1  
69 0067 1 LIBRARY 'NXPORT:XPORT'; ! XPORT Library  
70 0068 1 REQUIRE 'REQ:RNODEF'; ! RUNOFF variant definitions  
71 0199 1  
72 U 0200 1 %IF DSRPLUS %THEN  
73 U 0201 1 LIBRARY 'REQ:DPLLIB'; ! DSRPLUS BLISS Library  
74 0202 1 %ELSE  
75 0203 1 LIBRARY 'REQ:DSRLIB'; ! DSR BLISS Library  
76 0204 1 %FI  
77 0205 1  
78 0206 1 !  
79 0207 1 ! MACROS:  
80 0208 1 !  
81 0209 1 !  
82 0210 1 ! EQUATED SYMBOLS:  
83 0211 1 !  
84 0212 1 !  
85 0213 1 ! OWN STORAGE:  
86 0214 1 !  
87 0215 1 !  
88 0216 1 ! EXTERNAL REFERENCES:  
89 0217 1 !  
90 0218 1 !  
91 0219 1 EXTERNAL  
92 0220 1 FNCT : FNCT_DEFINITION,  
93 0221 1 FNESIZ : FN-EXT-SIZE_DÉFINITION,  
94 0222 1 FNISIZ : FN-INT-SIZE_DEFINITION,  
95 0223 1 HCT : HCT_DEFINITION,  
96 0224 1 PHAN : PHAN_DEFINITION;
```

```
98 0225 1 %SBTTL 'Routine TPR'
99 0226 1 GLOBAL ROUTINE TPR (COUNT) =
100 0227 1
101 0228 1 ++
102 0229 1 FUNCTIONAL DESCRIPTION:
103 0230 1
104 0231 1 TPR is used to see if COUNT lines are still available in
105 0232 1 the text portion of a page. The text portion include footnotes.
106 0233 1
107 0234 1 FORMAL PARAMETERS:
108 0235 1
109 0236 1 COUNT indicates how many lines are to be available.
110 0237 1
111 0238 1 IMPLICIT INPUTS: None
112 0239 1
113 0240 1 IMPLICIT OUTPUTS: None
114 0241 1
115 0242 1 ROUTINE VALUE:
116 0243 1 COMPLETION CODES:
117 0244 1
118 0245 1 Returns TRUE if the specified number of lines are available,
119 0246 1 otherwise returns FALSE.
120 0247 1
121 0248 1 SIDE EFFECTS: None
122 0249 1
123 0250 1 --
124 0251 1
125 0252 2 BEGIN
126 0253 2
127 0254 2 IF .phan_top_page
128 0255 2 OR (NOT .phan_paging)
129 0256 2 THEN
130 0257 2 RETURN true;
131 0258 2
132 0259 2 IF (.count + .phan_lines_tp + .hct_layoutn) GTR .phan_llines
133 0260 2 THEN
134 0261 2 RETURN false
135 0262 2 ELSE
136 0263 2 RETURN true
137 0264 2
138 0265 1 END;
```

```

.TITLE TPR
.IDENT \V04-000\

.EXTRN FNCT, FNESIZ, FNISIZ
.EXTRN HCT, PHAN

.PSECT $CODE$, NOWRT, 2

.ENTRY TPR, Save R2
MOVAB PHAN, R2
BLBS PHAN, 1$  

BLBC @PHAN+40, 1$  

ADDL3 PHAN+12, COUNT, R0
ADDL2 HCT+32, R0

```

			0004	00000	.ENTRY	TPR, Save R2
		52 00000000G	EF	9E 00002	MOVAB	PHAN, R2
		1A	62	E8 00009	BLBS	PHAN, 1\$
		16	28	B2 E9 0000C	BLBC	2PHAN+40, 1\$
	04	AC	0C	A2 C1 00010	ADDL3	PHAN+12, COUNT, R0
50		50 00000000G	EF	C0 00016	ADDL2	HCT+32, R0

TPR
V04-000

Routine TPR

M 1
16-Sep-1984 01:55:26
14-Sep-1984 13:08:21 VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]TPR.BLI;1 Page 5
(4)

04 B2	50 D1 0001D 03 15 00021 50 D4 00023 50 04 00025 50 01 D0 00026 1\$: 04 00029	CMPL R0, @PHAN+4 BLEQ 1\$ CLRL R0 RET MOVL #1, R0 RET	
-------	---	--	--

0263
0265

; Routine Size: 42 bytes, Routine Base: \$CODE\$ + 0000

TPR
V04

```

: 140 0266 1 XSBTTL 'Routine TPBEQL'
: 141 0267 1 GLOBAL ROUTINE TPBEQL (COUNT) =
: 142 0268 1
: 143 0269 1 ++
: 144 0270 1 FUNCTIONAL DESCRIPTION:
: 145 0271 1
: 146 0272 1 TPBEQL is used to see if exactly COUNT lines are still available in
: 147 0273 1 the text portion of a page. The text portion include footnotes.
: 148 0274 1
: 149 0275 1 FORMAL PARAMETERS:
: 150 0276 1
: 151 0277 1 COUNT indicates how many lines are to be available.
: 152 0278 1
: 153 0279 1 IMPLICIT INPUTS: None
: 154 0280 1
: 155 0281 1 IMPLICIT OUTPUTS: None
: 156 0282 1
: 157 0283 1 ROUTINE VALUE:
: 158 0284 1 COMPLETION CODES:
: 159 0285 1
: 160 0286 1 Returns TRUE if exactly the specified number of lines are available,
: 161 0287 1 otherwise returns FALSE.
: 162 0288 1
: 163 0289 1 SIDE EFFECTS: None
: 164 0290 1
: 165 0291 1 --
: 166 0292 1
: 167 0293 2 BEGIN
: 168 0294 2
: 169 0295 2 IF (.count + .phan_lines_tp + .hct_layoutn) NEQ .phan_llines
: 170 0296 2 THEN
: 171 0297 2 RETURN false
: 172 0298 2 ELSE
: 173 0299 2 RETURN true
: 174 0300 2
: 175 0301 1 END;                                !End of TPBEQL

```

50	04	AC	00000070G	0000 00000	.ENTRY	TPBEQL, Save nothing	0267
			50 00000000G	EF C1 00002	ADDL3	PHAN+12, COUNT, R0	0295
			00000000G FF	EF C0 0000B	ADDL2	HCT+32, R0	
				50 D1 00012	CMPL	R0, @PHAN+4	
				03 13 00019	BEQL	1\$	
				50 D4 0001B	CLRL	R0	
				04 0001D	RET		0299
				50 0001E 1\$:	MOVL	#1, R0	
				04 00021	RET		0301

; Routine Size: 34 bytes, Routine Base: \$CODES + 002A

```
177      0302 1 %SBTTL 'Routine TPFEQL'
178      0303 1 GLOBAL ROUTINE TPFEQL =
179      0304 1
180      0305 1 !++
181      0306 1 FUNCTIONAL DESCRIPTION:
182      0307 1
183      0308 1 TPFEQL is used to see if exactly enough space is available in the
184      0309 1 text portion of the page to fit one or more footnotes there.
185      0310 1
186      0311 1 FORMAL PARAMETERS: None
187      0312 1
188      0313 1 IMPLICIT INPUTS: None
189      0314 1
190      0315 1 IMPLICIT OUTPUTS: None
191      0316 1
192      0317 1 ROUTINE VALUE:
193      0318 1 COMPLETION CODES:
194      0319 1
195      0320 1 Returns the number of footnotes for which exactly enough room is
196      0321 1 available. Returns 0 if no footnotes will fit.
197      0322 1
198      0323 1 SIDE EFFECTS: None
199      0324 1
200      0325 1 !--
201      0326 1
202      0327 2 BEGIN
203      0328 2
204      0329 2 LOCAL
205      0330 2     total_fit_size;
206      0331 2
207      0332 2     total_fit_size = 0;           !Don't know if any footnotes will fit yet.
208      0333 2
209      0334 2 !Now, loop through the list of footnotes that are elegible to go out.
210      0335 2 !Quit either when you run out of footnotes to look at, or you run out
211      0336 2 !of footnotes that will fit.
212      0337 2 INCR i FROM 1 TO .fnct_ready DO
213      0338 3 BEGIN
214      0339 3     !First check to see if there is enough room at all for this footnote.
215      0340 3     IF
216      0341 4       .phan_llines geq (.total_fit_size + .fnesiz [.i - 1 + .fnct_old] + .phan_lines_tp + .hct_layoutn
217      0342 3 THEN
218      0343 3     !Ok, we know there's a chance. Now check to see if we're exactly at the
219      0344 3     !right spot.
220      0345 3     IF
221      0346 3       tpbeql (.total_fit_size + .fnesiz [.i - 1 + .fnct_old])
222      0347 3 THEN
223      0348 3     !Found the exact position where some footnotes can be output
224      0349 4     BEGIN
225      0350 4     !Ideally, we could just exit with the value of I indicating the
226      0351 4     !number of footnotes that will exactly fit. However, there is a
227      0352 4     !strange case in which the footnote generates no text. For example,
228      0353 4     !the user might have input a footnote containing only indexing commands.
229      0354 4     !The following adjusts for that, if that is the case.
230      0355 4     INCR j FROM .I TO .fnct_ready - 1 DO
231      0356 4     IF .fnesiz [.j + .fnct_old] EQL 0 !Look ahead one more footnote
232      0357 4     THEN
233      0358 4     !Ok, the next footnote is a zero-length one.
```

```

234      0359  4          i = .i + 1
235      0360  4          ELSE EXITLOOP;
236      0361  4
237      0362  4
238      0363  4          !Now, finally, we've included any zero-length footnotes.
239      0364  4          RETURN .i
240      0365  4          END
241      0366  3          ELSE
242      0367  3          !Didn't fit exactly, but still fit. Add its size to the total
243      0368  3          !size of footnotes that will fit so far at this spot.
244      0369  4          BEGIN
245      0370  4          total_fit_size = .total_fit_size + .fnesiz [.i - 1 + .fnct_old];
246      0371  4          END
247      0372  3          ELSE
248      0373  3          !Nothing fits. Some previous footnotes may have fitted, but this one
249      0374  3          is just a bit too large.
250      0375  3          RETURN 0
251      0376  2          END;
252      0377  2
253      0378  2          !Falling through the loop means either no footnotes at all were found,
254      0379  2          or, we were not exactly at the correct position for outputting at
255      0380  2          least one footnote, even though there might be footnotes ready.
256      0381  2          RETURN 0
257      0382  2
258      0383  1          END;

```

!End of TPFEQL

			007C 00000	.ENTRY	TPFEQL, Save R2,R3,R4,R5,R6	0303
	56	00000000G	EF 9E 00002	MOVAB	FNESIZ-4, R6	
	55	00000000G	EF 9E 00009	MOVAB	FNCT+12, R5	
	54	F8	A5 D0 00010	MOVL	FNCT+4, R4	0337
			52 7C 00014	CLRQ	I	
			50 11 00016	BRB	6\$	
50	52		65 C1 00018 1\$: 6640 D0 0001C	ADDL3	FNCT+12, I, R0	0341
	50			MOVL	FNESIZ-4[R0], R0	
51	53		50 C1 00020	ADDL3	R0, TOTAL FIT SIZE, R1	
50	51	00000000G	EF C1 00024	ADDL3	PHAN+12, R1, R0	
	50	00000000G	EF C0 0002C	ADDL2	HCT+32, R0	
	50	00000000G	FF D1 00033	CMPL	BPHAN+4, R0	
			30 19 0003A	BLSS	7\$	
			51 DD 0003C	PUSHL	R1	0346
9C	AF		01 FB 0003E	CALLS	#1. TPBEQL	
	18		50 E9 00042	BLBC	R0, 5\$	
	50	FF	A2 9E 00045	MOVAB	-1(R2), J	0356
			0C 11 00049	BRB	3\$	
51	50		65 C1 0004B 2\$: 04 A641 D5 0004F	ADDL3	FNCT+12, J, R1	
			07 12 00053	TSTL	FNESIZ[R1]	
			52 D6 00055	BNEQ	4\$	
EF	50	F8	A5 F2 00057 3\$: 50 52 D0 0005C 4\$: 04 0005F	INCL	I	0359
				AOBLSS	FNCT+4, J, 2\$	0356
				MOVL	I, R0	0364
50	52		65 C1 00060 5\$: 6640 C0 00064	RET		
	53			ADDL3	FNCT+12, I, R0	0370
				ADDL2	FNESIZ-4[R0], TOTAL_FIT_SIZE	

TPR
V04-000

Routine TPFEQL

D 2
16-Sep-1984 01:55:26
14-Sep-1984 13:08:21

VAX-11 BLISS-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]

Page 9
1 (6)

1

52

54 F3 00068 6S: AOBLEQ R4. I. 18
50 D4 0006C 7S: CLRL RO
04 0006E RET

0338
0383

; Routine Size: 111 bytes. Routine Base: \$CODES + 004C

```
0384 1 %SBTTL 'Routine TPFFIT'  
0385 1 GLOBAL ROUTINE TPFFIT =  
0386 1 !++  
0388 1 : FUNCTIONAL DESCRIPTION:  
0389 1 :  
0390 1 : Checks to see how many footnotes could be output in the space remaining on the page.  
0391 1 : This routine doesn't check as closely as TPFEQL: it's ok for extra space to be available.  
0392 1 :  
0393 1 : FORMAL PARAMETERS: None  
0394 1 :  
0395 1 : IMPLICIT INPUTS: None  
0396 1 :  
0397 1 : IMPLICIT OUTPUTS: None  
0398 1 :  
0399 1 : ROUTINE VALUE:  
0400 1 : COMPLETION CODES:  
0401 1 :  
0402 1 : Returns the number of footnotes for which sufficient space is available on  
0403 1 : the current page.  
0404 1 :  
0405 1 : SIDE EFFECTS: None  
0406 1 :  
0407 1 :--  
0408 1 :  
0409 2 : BEGIN  
0410 2 :  
0411 2 : LOCAL  
0412 2 :     total_fit_size;  
0413 2 :  
0414 2 :     total_fit_size = 0;           !Don't know if any footnotes will fit yet.  
0415 2 :  
0416 2 : !Loop through the list of read footnotes, and quit when you find the first one  
0417 2 : !that won't fit anymore.  
0418 2 : INCR i FROM 1 TO .fnct_ready DO  
0419 2 :  
0420 2 : !NOTE: The following logical expression is almost equivalent to  
0421 2 : TPR (.TOTAL_FIT_SIZE + .FNESIZ [i - 1 + .FNCT_OLD]) except that  
0422 2 : you can't use TPR when checking for space for Footnotes. That's  
0423 2 : because when it's time to check for footnote space, PHAN_TOP_PAGE  
0424 2 : is TRUE, and when that's the case TPR always returns TRUE, which  
0425 2 : would ultimately result in all ready footnotes being output, even  
0426 2 : if there's not enough space.  
0427 2 : IF .phan_llines GEQ (.total_fit_size + .fnesiz [i - 1 + .fnct_old] + .phan_lines_tp + .hct_layoutn)  
0428 2 : THEN  
0429 2 :     total_fit_size = .total_fit_size + .fnesiz [i - 1 + .fnct_old]    !Add size to running total.  
0430 2 : ELSE  
0431 2 :     RETURN .i - 1;          !The last footnote looked at is the last one that will fit.  
0432 2 :  
0433 2 : !Falling through the loop means that all the footnotes will fit.  
0434 2 : RETURN .fnct_ready  
0435 2 :  
0436 1 : END;                  !End of TPFFIT
```

TPR
V04-000

Routine TPFFIT

F 2
16-Sep-1984 01:55:26
14-Sep-1984 13:08:21VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]TPR.BLI;1Page 11
(7)

UNF

		54 00000000G	001C 00000	.ENTRY	TPFFIT, Save R2,R3,R4	: 0385
			EF 9E 00002	MOVAB	FNCT+4, R4	: 0414
			53 D4 00009	CLRL	TOTAL_FIT_SIZE	: 0427
			50 D4 00008	CLRL	I	
			35 11 0000D	BRB	3S	
51		50 00000000G	08 A4 C1 0000F	1\$: ADDL3	FNCT+12, I, R1	
51			EF 41 D0 00014	MOVL	FNESIZ-4[R1], R2	
		52 00000000GEF	S2 C1 0001C	ADDL3	R2, TOTAL_FIT_SIZE, R1	
		51 00000000G	EF C0 00020	ADDL2	PHAN+12, R1	
		51 00000000G	EF C0 00027	ADDL2	HCT+32, R1	
		51 00000000G	FF D1 0002E	CMPL	@PHAN+4, R1	
			05 19 00035	BLSS	2S	
			52 C0 00037	ADDL2	R2, TOTAL_FIT_SIZE	: 0429
			08 11 0003A	BRB	3S	
	51	FF	A0 9E 0003C	2\$: MOVAB	-1(R0), R1	: 0431
	50		51 D0 00040	MOVL	R1, R0	
			04 00043	RET		
C7		50	64 F3 00044	3\$: AOBLEQ	FNCT+4, I, 1S	: 0427
		50	64 D0 00048	MOVL	FNCT+4, R0	: 0434
			04 0004B	RET		: 0436

: Routine Size: 76 bytes, Routine Base: \$CODES + 00BB

Routine TPFSKP

```

314 0437 1 %SBTTL 'Routine TPFSKP'
315 0438 1 GLOBAL ROUTINE TPFSKP (FOOTNOTE_COUNT) =
316 0439 1
317 0440 1 ++
318 0441 1 FUNCTIONAL DESCRIPTION:
319 0442 1
320 0443 1 Computes how many lines need to be skipped in order to be precisely
321 0444 1 at the position for outputting a certain number of footnotes.
322 0445 1
323 0446 1 FORMAL PARAMETERS:
324 0447 1
325 0448 1 FOOTNOTE_COUNT indicates how many of the ready footnotes are to be output.
326 0449 1
327 0450 1 IMPLICIT INPUTS: None
328 0451 1
329 0452 1 IMPLICIT OUTPUTS: None
330 0453 1
331 0454 1 ROUTINE VALUE:
332 0455 1 COMPLETION CODES: None
333 0456 1
334 0457 1 SIDE EFFECTS: None
335 0458 1
336 0459 1 --
337 0460 1
338 0461 2 BEGIN
339 0462 2
340 0463 2 LOCAL
341 0464 2     total_fit_size;
342 0465 2
343 0466 2     total_fit_size = 0;
344 0467 2     !Add up sizes of the specified number of footnotes.
345 0468 2     INCR i FROM 1 TO .footnote_count DO
346 0469 2         !(Forget old footnotes.)
347 0470 2         total_fit_size = .total_fit_size + .fnesiz [.i - 1 + .fnct_old];
348 0471 2
349 0472 3 RETURN (.phan_llines - (.total_fit_size + .phan_lines_tp + .hct_layoutn))
350 0473 3
351 0474 1 END:                                !End of TPFSKP

```

		0004 00000	.ENTRY	TPFSKP, Save R2	: 0438
		50 7C 00002	CLRQ	I	: 0470
		10 11 00004	BRB	2\$	
52	50 00000000G	EF C1 00006	ADDL3	FNCT+12, I, R2	
	51 00000000GEF	42 C0 0000E	ADDL2	FNESIZ-4[R2], TOTAL_FIT_SIZE	
EB	50 04	AC F3 00016	AOBLEQ	FOOTNOTE_COUNT, I, TS	
	51 00000000G	EF C0 0001B	ADDL2	PHAN+12, R1	: 0472
	51 00000000G	EF C0 00022	ADDL2	HCT+32, R1	
	FF	51 C3 00029	SUBL3	R1, @PHAN+4, R0	
		04 00031	RET		: 0474

: Routine Size: 50 bytes. Routine Base: \$CODE\$ + 0107

```

353 0475 1 XSBTTL 'Routine TPFSIZ'
354 0476 1 GLOBAL ROUTINE TPFSIZ (FOOTNOTE_COUNT) =
355 0477 1
356 0478 1 !++
357 0479 1 FUNCTIONAL DESCRIPTION:
358 0480 1
359 0481 1 Computes how many lines a specified number of footnotes occupies.
360 0482 1
361 0483 1 FORMAL PARAMETERS:
362 0484 1
363 0485 1 FOOTNOTE_COUNT indicates how many of the ready footnotes are to be counted.
364 0486 1
365 0487 1 IMPLICIT INPUTS: None
366 0488 1
367 0489 1 IMPLICIT OUTPUTS: None
368 0490 1
369 0491 1 ROUTINE VALUE:
370 0492 1 COMPLETION CODES:
371 0493 1
372 0494 1 Returns the number of lines that the footnotes will take up.
373 0495 1
374 0496 1 SIDE EFFECTS: None
375 0497 1
376 0498 1 !--
377 0499 1
378 0500 2 BEGIN
379 0501 2
380 0502 2 LOCAL
381 0503 2     total_fit_size;
382 0504 2
383 0505 2     total_fit_size = 0;
384 0506 2     !Add up sizes of the specified number of footnotes.
385 0507 2     INCR i FROM 1 TO .footnote_count DO
386 0508 2         !(Forget old footnotes.)
387 0509 2         total_fit_size = .total_fit_size + .fnesiz [.i - 1 + .fnct_old];
388 0510 2
389 0511 2 RETURN .total_fit_size
390 0512 2
391 0513 1 END;
                                         !End of TPFSIZ

```

		0004 00000	.ENTRY	TPFSIZ, Save R2	: 0476
		52 D4 00002	CLRL	TOTAL_FIT_SIZE	: 0505
		50 D4 00004	CLRL	I	: 0509
		10 11 00006	BRB	2\$	
51	50 00000000G	EF C1 00008 1\$:	ADDL3	FNCT+12, I, R1	
	52 00000000GEF41	C0 00010	ADDL2	FNESIZ-4[R1], TOTAL_FIT_SIZE	
EB	50 04 AC	F3 00018 2\$:	AOBLEQ	FOOTNOTE COUNT, I, TS	
	52 D0 0001D	04 00020	MOVL	TOTAL_FIT_SIZE, R0	: 0511
			RET		: 0513

; Routine Size: 33 bytes. Routine Base: \$CODE\$ + 0139

TPR
V04-000

Routine TPFSIZ

I 2
16-Sep-1984 01:55:26 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 13:08:21 DISK\$VMSMASTER:[RUNOFF.SRC]TPR.BLI;1 Page 14
(9)

: 392 0514 1
: 393 0515 1 END
: 394 0516 0 ELUDOM

!End of module

UNP
V04

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	346	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	17	1	86	00:00.3

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:TPR/OBJ=OBJ\$:TPR MSRC\$:TPR/UPDATE=(ENH\$:TPR)

Size: 346 code + 0 data bytes
Run Time: 00:07.8
Elapsed Time: 00:17.7
Lines/CPU Min: 3948
Lexemes/CPU-Min: 14640
Memory Used: 48 pages
Compilation Complete

0350 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

